UNDECIDABILITY OF RECOGNIZING AXIOMATIZATIONS FOR DEDUCTIVE PROPOSITIONAL CALCULI

GRIGORIY V. BOKOV

Abstract. In this paper we consider deductive propositional calculi, that are finitely axiomatizable extensions of intuitionistic implicational propositional calculus together with the rules of modus ponens and substitution. We give a proof of undecidability of recognizing axiomatizations for each deductive propositional calculus. Moreover, the problems of recognizing extensions and completeness are also undecidable for these calculi. Particularly, this holds for each superintuitionistic calculus.

§1. Introduction. In general, a propositional calculus to represent a finite set of propositional formulas over some signature together with a finite set of rules of inferences. The recognizing axiomatizations for a propositional calculus is the following problem: whether a given finite set of propositional formulas constitutes (axiomatizes) an adequate axiom system for this calculus, i.e. each formula of a calculus is derivable from a given set of formulas by rules of calculus. The question of decidability of this problem were proposed by Tarski in 1946 [13]. In this paper we consider only the classical propositional calculus with the rules of modus ponens and substitution.

The undecidability of recognizing axiomatizations for classical propositional calculus was obtained due to Linial and Post in 1949 [7]. They gave sketch of proofs of a number of results, one of them expressible in the form that it is undecidable whether a given finite set of propositional formulas axiomatizes all classical tautologies. Note that it was considered only formulas over the signature $\{\neg, \vee\}$ and the rule of modus ponens was formulated appropriately. Later the proof of their result was restored by Davis [2, pp. 137–142] and a complete proof was appeared in the work of Yntema [16].

For the intuitionistic propositional calculus over the signature $\{\neg, \vee, \&, \to\}$ same result was proved Kuznetsov in 1963 [6]. Moreover, he proved that this holds for each superintuitionistic calculus, i.e. a finitely axiomatizable extension of the intuitionistic propositional calculus. Particularly, this holds for classical propositional calculus and proves the Linial and Post theorem.

In 1961 A. A. Markov (Jr.) proposes the following problem: is it decidable whether a given finite set of implicational propositional formulas, i.e. formulas over the signature $\{\to\}$, axiomatizes all classical implicational tautologies? Kuznetsov in [6] mentioned that this problem seems still to be open.

Key words and phrases. Deductive propositional calculi, classical and intuitionistic propositional calculi, implicational calculus, finite axiomatization, tag system.

In 1994 Marcinkowski [9] prove that the Markov's problem is undecidable. Moreover, Marcinkowski obtained a much stronger result: fix a implicational propositional tautology A that is not of the form $B \to B$ for some formula B, it is undecidable whether A derivable from a given finite set of implicational formulas by rules of modus ponens and substitution.

Recently, Zolin in 2013 [18] prove the result of Kuznetsov for the superintuitionistic propositional calculus over signatures $\{\wedge, \to\}$ and $\{\vee, \to\}$. It is based on the so-called tag systems introduced by Post [12] and proposed in 2009 by Bokov [1] for a proof of the result of Linial and Post. Besides Zolin in [18] give a detailed and useful historical survey of related results.

The aim of this paper is to prove the undecidability of recognizing axiomatizations, extensions and completeness for each deductive propositional calculus over a signature containing the connective $\to$. The deductive propositional calculus is a finitely axiomatizable extension of intuitionistic implicational propositional calculus.

§2. Preliminaries and results. Let $\mathcal{V}$ be a infinite set of propositional variables. Letters x, y, z, u, etc., are used to denote propositional variables. The signature Σ is a finite set of connectives. Each connective attached to unique, classical, two-valued truth-function, but different connectives may represent the same truth-function. Usually connectives are binary or unary such as $\{\neg, \vee, \wedge, \to\}$.

Propositional formulas constructs from the signature Σ (or Σ-formulas) and propositional variables $\mathcal{V}$ in the usual way. Capital letters A, B, C, etc., are used for propositional formulas. Throughout the paper, we will omit the outermost parentheses in formulas and parentheses assuming the customary priority of connectives.

In this paper we will consider arbitrary signatures containing the binary symbol $\to$. Note that by Gladstone [3] we can suppose that the signature Σ does not contain the symbol $\to$, but there is some propositional formula having x, y as sole variables, whose truth-table interpretation is "x implies y". In this case we denote the specified formula simply by $x \to y$.

A propositional calculus P over a signature Σ (or Σ-calculus) is a system consisting of a finite set P of Σ-formulas referred to as axiom and the two rules of inference:

1) modus ponens

$$A, A \to B \vdash B;$$

2) substitution

$$A \vdash \sigma A,$$

where σA is the substitution instance of A, i.e. the result of applying the substitution σ to the formula A.

The set of derivable (or provable) formulas of a calculus P denote by $[P]$. A derivation in P is defined from the axioms and rules of inference in the usual way. The statement that a formula A is drivable from P denote by $P \vdash A$.

Let us introduce the following partial order relation on the set of all propositional calculus. We say that a propositional calculus P_1 is weaker than a

propositional calculus P_2 and write this by $P_1 \leq P_2$ (or, equivalently, $P_2 \geq P_1$) if each drivable formula of P_1 is also drivable from P_2, i.e., if $[P_1] \subseteq [P_2]$. We write $P_1 \sim P_2$ and say that two calculi P_1 and P_2 are equivalent if $P_1 \leq P_2$ and $P_2 \leq P_1$. Finally, we write $P_1 < P_2$ if $P_1 \leq P_2$ and $P_1 \not\sim P_2$.

Denote by $\mathbf{Cl}_\Sigma$ the classical propositional calculus over a signature Σ, and by $\mathbf{Int}_\Sigma$ the intuitionistic propositional calculus over a signature Σ [5]. Since many tautologies of $\mathbf{Cl}_\Sigma$ are not provable within $\mathbf{Int}_\Sigma$, e.g. $x \to y \sim \overline{x} \vee y$, we suppose that the signature Σ of the intuitionistic propositional calculus $\mathbf{Int}_\Sigma$ contains the binary connective $\to$.

Consider the intuitionistic implicational propositional calculus $\mathbf{Int}_\to$. As well known [4] and [17, p.8], an axiom system of $\mathbf{Int}_\to$ is given by Łukasiewicz as follows:

$$\begin{array}{ll} (\mathrm{A}_1) & x \to (y \to x), \\ (\mathrm{A}_2) & (x \to (y \to z)) \to ((x \to y) \to (x \to z)). \end{array}$$

The classical implicational propositional calculus $\mathbf{Cl}_\to$ obtained from $\mathbf{Int}_\to$ by adding the Peirce's law [14, p.52]:

$$\begin{array}{ll} (\mathrm{A}_1) & x \to (y \to x), \\ (\mathrm{A}_2) & (x \to (y \to z)) \to ((x \to y) \to (x \to z)), \\ (\mathrm{Peirce}) & ((x \to y) \to x) \to x. \end{array}$$

Now we define few recognizing problems for a fixed propositional calculus P_0.

Problem (Recognizing axiomatizations). Given a propositional calculus P, determine whether $P_0 \sim P$.

Problem (Recognizing extensions). Given a propositional calculus P, determine whether $P_0 \leq P$.

Problem (Recognizing completeness). Given a propositional calculus P such that $P \leq P_0$, determine whether $P_0 \leq P$.

Previous results can be summarized as follows.

Theorem 2.1 (Linial and Post, 1949). *The problems of recognizing axiomatizations, extensions, completeness for* $\mathbf{Cl}_{\{\neg,\vee\}}$ *are undecidable.*

Theorem 2.2 (Kuznetsov, 1963). *Fix a calculus* $P_0 \geq \mathbf{Int}_{\{\neg,\vee,\&,\to\}}$*, the problems of recognizing axiomatizations, extensions, completeness for* P_0 *are undecidable.*

Theorem 2.3 (Marcinkowski, 1994). *Fix a* $\{\to\}$*-tautology* A *that is not of the form* $B \to B$ *for some formula* B*, the problem of recognizing extensions for a* $\{\to\}$*-calculus* $\{A \to A\}$ *is undecidable.*

Since the implicational calculi $\mathbf{Cl}_{\{\to\}}$ and $\mathbf{Int}_{\{\to\}}$ can be axiomatized by the single formulas, as shown by Łukasiewicz [8] and Meredith [10],

$$\mathbf{Cl}_{\{\to\}} \sim \{((x \to y) \to z) \to ((z \to x) \to (u \to x))\}$$
$$\mathbf{Inv}_{\{\to\}} \sim \{((x \to y) \to z) \to (u \to ((y \to (z \to v)) \to (y \to v)))\}$$

the following result also makes sense.

COROLLARY 2.4. *The problems of recognizing axiomatizations, extensions, completeness for* $\mathbf{Cl}_{\{\to\}}$ *and the problem of recognizing extensions for* $\mathbf{Int}_{\{\to\}}$ *are undecidable.*

THEOREM 2.5 (Zolin, 2013). *Fix a signature* $\Sigma \supseteq \{\wedge, \to\}$ *and a calculus* $P_0 \geq \mathbf{Int}_{\{\wedge,\to\}}$, *the problems of recognizing axiomatizations, extensions, completeness for* P_0 *are undecidable.*

Let us call a propositional calculus P *deductive* if $P \geq \mathbf{Inv}_{\{\to\}}$. For example, the classical propositional calculus $\mathbf{Cl}_{\{\neg,\vee,\wedge,\to\}}$, the intuitionistic propositional calculus $\mathbf{Int}_{\{\neg,\vee,\wedge,\to\}}$ and the inconsistent propositional calculus $\{x\}$ are weak-deductive calculi. The latter calculus is inconsistent, since all formulas are derivable from it by the rule of substitution. Our main result is the following theorems.

THEOREM 2.6. *Fix a signature* $\Sigma \supseteq \{\to\}$, *the problem of recognizing extensions of* P_0 *is undecidable for all* Σ*-calculus* P_0.

THEOREM 2.7. *Fix a signature* $\Sigma \supseteq \{\to\}$, *the problem of recognizing completeness of* P_0 *is undecidable for all deductive* Σ*-calculus* P_0.

As corollary we have the undecidability of problem of recognizing axiomatizations.

COROLLARY 2.8. *Fix a signature* $\Sigma \supseteq \{\to\}$, *the problem of recognizing axiomatizations of* P_0 *is undecidable for all deductive* Σ*-calculus* P_0.

Moreover, if we take in the theorem 2.6 $P_0 = \{A\}$ for a Σ-formula A, so we have the undecidability of problem of derivability.

COROLLARY 2.9. *Fix a signature* $\Sigma \supseteq \{\to\}$ *and a* Σ*-formula* A, *the following problem is undecidable:*

$$\text{given a } \Sigma\text{-calculus } P, \text{ determine whether } P \vdash A.$$

Particularly, this holds for a formula A of the form $B \to B$ for some formula B in contrast with the theorem 2.3.

§3. The proof of undecidability. In order to prove the Theorem 2.6, we shall effectively reduce the halting problem of tag systems to the problem of recognizing axiomatizations of deductive propositional calculi. Then, the proof of Theorem 2.6 is immediate from the undecidability of the halting problem [11].

More precisely, we fix any signature Σ such that $\{\to\} \subseteq \Sigma$ and any Σ-calculus P_0. For given a tag system T and a word ω we will construct a Σ-calculus $P = P_{T,\omega,P_0}$ such that T halts on the input word ω iff $P_0 \leq P$.

For the proof the Theorem 2.7 we will show that $P_{T,\omega,P_0} \leq P_0$ for every deductive Σ-calculus P_0.

First let us recall the notion of tag system introduced by Post [12].

3.1. Tag systems. Let $\mathcal{A}$ be a finite alphabet of letters $a_1, \ldots, a_m$. By $\mathcal{A}^*$ denote the set all words on $\mathcal{A}$, including the empty word. Let $\alpha \in \mathcal{A}^*$; $|\alpha|$ denotes the length of a word α.

Definition 3.1 (Post, [12]). A tag system is a triple $T = \langle \mathcal{A}, \mathcal{W}, d \rangle$, where $\mathcal{A} = \{a_1, \ldots, a_m\}$ is a finite alphabet of m symbols, $\mathcal{W} = \{\omega_1, \ldots, \omega_m\} \subseteq \mathcal{A}^*$ is a set of m words, and $d \in \mathbb{N}$ is a deletion number. Each of words $\omega_1, \ldots, \omega_m$ corresponds with one of the letters from the alphabet $\mathcal{A}$: $a_1 \to \omega_1, \ldots, a_m \to \omega_m$.

We say that T is applicable to a word $\alpha \in \mathcal{A}^*$ if $|\alpha| \geq d$. The application of T to a word $\alpha \in \mathcal{A}^*$ is defined as follows. Examine the first letter of the word α. If it is a_i then

1. remove from α the first d letters, and
2. append to its end the word ω_i.

Perform the same operation on the resulting word, and repeat the process so long as the resulting word has d or more letters. To be precise, if $\alpha = a_i\beta\gamma$, $|\beta| = d - 1$, and $\gamma \in \mathcal{A}^*$, then T produces the word $\gamma\omega_i$ from the word $a_i\beta\gamma$. Denote this production by $a_i\beta\gamma \overset{T}{\longmapsto} \gamma\omega_i$. We write $\alpha \overset{T}{\Longmapsto} \beta$ if there are words $\gamma_1, \ldots, \gamma_n$, $n \geq 1$, such that $\alpha = \gamma_1$, $\beta = \gamma_n$, and $\gamma_i \overset{T}{\longmapsto} \gamma_{i+1}$ for all $1 \leq i \leq n-1$.

Define the halting problem of tag systems. We say that a tag system T halts on a word $\alpha \in \mathcal{A}^*$ if there exists a word $\beta \in \mathcal{A}^*$ such that $\alpha \overset{T}{\Longmapsto} \beta$ and T is not applicable to β, i.e. $|\beta| < d$. The halting problem, given a tag sustem $T = \langle \mathcal{A}, \mathcal{W}, d \rangle$, is to decide, for any word $\alpha \in \mathcal{A}^*$, whether T halts on α.

Theorem 3.2 (Minsky, [11]). *There is a tag system T for which the halting problem is undecidable.*

Moreover, Wang [15] showed that this holds even for some tag system T with $d = 2$ and $1 \leq |\omega_i| \leq 3$ for all $1 \leq i \leq m$. For this reason, further we will assume that all words ω_i are nonempty.

3.2. Encoding letters and words. Let $\mathcal{A}$ be a finite set $\{a_1, \ldots, a_m\}$. We encode letters and words on $\mathcal{A}$ as $\{\to\}$-formulas.

Fix x^0 be a variable not occurring in P_0. Then the code of the letter $a_i \in \mathcal{A}$, for $1 \leq i \leq m$, is a formula

$$\overline{a_i} := ((x^0 \to \underbrace{x^0) \to \cdots \to x^0)}_{i} \to (x^0 \to (x^0 \to x^0)).$$

It is easily shown that

$$\mathbf{Inv}_{\{\to\}} \vdash z \to A$$

whenever a formula A is derivable from $\mathbf{Inv}_{\{\to\}}$. Since $x^0 \to (x^0 \to x^0)$ is the substitution instance of $x \to (y \to x)$, then we have the following lemma.

Lemma 3.3. $\mathbf{Inv}_{\{\to\}} \vdash \overline{a}$*, for every letter* $a \in \mathcal{A}$.

Now we introduce the following notation. Let $x \vee y$ be an abbreviation for $(x \to y) \to y$. For a nonempty word $\alpha = a_{i_1} \ldots a_{i_k} \in \mathcal{A}^*$, we write $\overrightarrow{\alpha}$ as a shortcut for the formula

$$\overline{a_{i_1}} \vee \left(\overline{a_{i_2}} \vee \cdots \vee \left(\overline{a_{i_{k-1}}} \vee \overline{a_{i_k}}\right)\right),$$

and $\overleftarrow{\alpha}$ as a shortcut for the formula

$$\left((\overline{a_{i_1}} \vee \overline{a_{i_2}}) \vee \cdots \vee \overline{a_{i_{k-1}}}\right) \vee \overline{a_{i_k}}.$$

The notations naturally extends to the alphabet $\mathcal{A} \cup \mathcal{V}$, where $\mathcal{V}$ is the infinite set of propositional variables defined above. For example, $\overrightarrow{axby} = \overline{a} \vee \big(x \vee \big(\overline{b} \vee y\big)\big)$, where $a, b \in \mathcal{A}$ and $x, y \in \mathcal{V}$.

LEMMA 3.4. *In* $\mathbf{Inv}_{\{\to\}}$ *the following derivations holds:*

$$\mathbf{Inv}_{\{\to\}} \vdash x \to x \vee y,$$
$$\mathbf{Inv}_{\{\to\}} \vdash y \to x \vee y.$$

DEFINITION 3.5. (Zolin, [18]) An alphabetic formula over the alphabet $\mathcal{A}$ or an $\mathcal{A}$-formula is an arbitrary $\{\vee\}$-formula over the codes of letters from $\mathcal{A}$. Formally, $\overline{a}$ is a $\mathcal{A}$-formula for each letter $a \in \mathcal{A}$, and if A, B are $\mathcal{A}$-formulas then so is $A \vee B$.

In particular, $\overrightarrow{\alpha}$ and $\overleftarrow{\alpha}$ are $\mathcal{A}$-formulas. By Lemma 3.3 a code of letter of $\mathcal{A}$ is derivable from $\mathbf{Inv}_{\{\to\}}$, therefore also every $\mathcal{A}$-formula is derivable from $\mathbf{Inv}_{\{\to\}}$ by Lemma 3.4.

LEMMA 3.6. $\mathbf{Inv}_{\{\to\}} \vdash A$, *for every* $\mathcal{A}$*-formula* A.

Given a formula A, denote by A^* the set of all substitution instances of A. Similarly, given a set M of formulas, by M^* denote the set

$$M^* := \bigcup_{A \in M} A^*.$$

In accordance with [18] let us call two formulas A and B unifiable if $A^* \cap B^* \neq \emptyset$.

LEMMA 3.7. *No two distinct* $\mathcal{A}$*-formulas are unifiable.*

PROOF. By induction on the definition of a $\mathcal{A}$-formula A.

Let A be the code of letter $a_i \in \mathcal{A}$. If B is the code of letter $a_j \in \mathcal{A}$, then $j \neq j$. Without loss of generality it can be assumed that $i < j$. Denote the following formula

$$((y \to \underbrace{x^0) \to \cdots \to x^0)}_{i} \to (x^0 \to (x^0 \to x^0))$$

by C. Since $\overline{a_i}$ is the substitution instances of C with respect to replacing the propositional variable y by x^0 and $\overline{a_j}$ is the substitution instances of C with respect to replacing the propositional variable y by

$$((x^0 \to \underbrace{x^0) \to \cdots \to x^0)}_{j-i},$$

we have that A and B are not unifiable.

If B is a formula $B_1 \vee B_2$ for some $\mathcal{A}$-formulas B_1 and B_2, then A is the substitution instances of

$$(y \to x^0) \to (x^0 \to (x^0 \to x^0))$$

and B is the substitution instances of $(u \to v) \to v$. Since formulas x^0 and $x^0 \to (x^0 \to x^0)$ are not unifiable, we see that also A and B are not unifiable.

Now let $A = A_1 \vee A_2$ for some $\mathcal{A}$-formulas A_1 and A_2, so it can be assumed that $B = B_1 \vee B_2$ for some $\mathcal{A}$-formulas B_1 and B_2. If A, B are unifiable, then

also A_1, B_1 and A_2, B_2 are unifiable. By the induction hypothesis, $A_1 = B_1$ and $A_2 = B_2$. Hence, $A = B$.

This completes the proof of the lemma. ⊣

By $\rhd$ denote the following formula

$$((x^0 \to x^0) \to x^0) \to x^0.$$

Since formulas $x \to x$ and $(y \to z) \to z$ are not unifiable, we obtain the following lemma.

LEMMA 3.8. *Formulas $\rhd$, $\rhd \to A$ are not unifiable for every formula A, and also formulas $\rhd$, $(\rhd \to B) \to C$ are not unifiable for all formulas B, C.*

Furthermore, we define a notion of the code of a word over $\mathcal{A}$. The code of a nonempty word $\alpha \in \mathcal{A}^*$ is the set $\mathsf{Code}(\alpha)$ consisting of tautologies of the following four types:

Type 0	$\rhd \to \overrightarrow{\alpha}$	
Type 1	$\rhd \to \overrightarrow{\alpha_1} \vee \overrightarrow{\alpha_2}$	$\alpha = \alpha_1\alpha_2$, $\lvert\alpha_1\rvert \geq 2$, $\lvert\alpha_2\rvert \geq 1$;
Type 2	$\rhd \to (\overleftarrow{\alpha_1} \vee \overrightarrow{\alpha_2}) \vee \overrightarrow{\alpha_3}$	$\alpha = \alpha_1\alpha_2\alpha_3$, $\lvert\alpha_1\rvert \geq 2$, $\lvert\alpha_2\rvert \geq 2$, $\lvert\alpha_3\rvert \geq 1$;
Type 3	$\rhd \to \overleftarrow{\alpha_1} \vee \overrightarrow{\alpha_2}$	$\alpha = \alpha_1\alpha_2$, $\lvert\alpha_1\rvert \geq 3$, $\lvert\alpha_2\rvert \geq 1$.

In addition, suppose that $\mathsf{Code}(\varepsilon) = \emptyset$. Furthermore, we will call each formula of $\mathsf{Code}(\alpha)$ as the code of same word α.

3.3. Construction of the calculus P_{T,ω,P_0}. Let $T = \langle \mathcal{A}, \mathcal{W}, d\rangle$ be a tag system, ω be a nonempty word on $\mathcal{A}$, and P_0 be a deductive Σ-calculus. Denote by P_{T,ω,P_0} a Σ-calculus with axioms:

(W_ω)	$\rhd \to \overrightarrow{\omega}$,	
(T_1)	$(\rhd \to \overrightarrow{a_i\alpha y}) \to (\rhd \to \overrightarrow{y\omega_i})$,	for all $\alpha \in \mathcal{A}^*$, $\lvert\alpha\rvert = d-1$, $1 \leq i \leq m$,
(T_2)	$(\rhd \to \overrightarrow{a_i\alpha}) \to (\rhd \to \overrightarrow{\omega_i})$,	for all $\alpha \in \mathcal{A}^*$, $\lvert\alpha\rvert = d-1$, $1 \leq i \leq m$,
(H)	$(\rhd \to \overrightarrow{\alpha}) \to A$,	for all $\alpha \in \mathcal{A}^*$, $0 < \lvert\alpha\rvert < d$, $A \in P_0$,
(R_1)	$(\rhd \to (y \vee \overrightarrow{az}) \vee u) \to (\rhd \to (\overleftarrow{ya} \vee z) \vee u)$,	for all $a \in \mathcal{A}$,
(R_2)	$(\rhd \to \overleftarrow{ya} \vee z) \to (\rhd \to y \vee \overrightarrow{az})$,	for all $a \in \mathcal{A}$.

Let P_T be a subsystem of P_{T,ω,P_0} consisting of axioms T_1, T_2, R_1, R_2 and $P_{T,\omega} = P_T \cup \mathrm{W}_\omega$. Now we prove some properties of the calculus P_{T,ω,P_0}.

LEMMA 3.9. $P_{T,\omega} \leq \mathbf{Inv}_{\{\to\}}$.

PROOF. Easily follows from Lemmas 3.3, 3.4 and 3.6. ⊣

COROLLARY 3.10. $P_{T,\omega,P_0} \leq P_0$.

3.4. Derivability of the T-productions. Here we show that the calculus P_T can "simulate" productions of the tag system T. At the beginning let us prove auxiliary lemmas.

LEMMA 3.11. $\mathrm{R}_1, \rhd \to (\overleftarrow{\xi} \vee \overrightarrow{\beta}) \vee \overrightarrow{\zeta} \vdash \rhd \to \overleftarrow{\xi\beta} \vee \overrightarrow{\zeta}$, *for all nonempty words $\xi, \beta, \zeta \in \mathcal{A}^*$.*

Proof. By induction on $|\beta|$. If $|\beta| = 1$, then the formulas $\rhd \to (\overleftarrow{\xi} \vee \overrightarrow{\beta}) \vee \overrightarrow{\zeta}$ and $\rhd \to \overleftarrow{\xi\beta} \vee \overrightarrow{\zeta}$ are identical.

Now let $|\beta| \geq 2$, then $\beta = a\delta$ for a letter $a \in \mathcal{A}$ and a nonempty word δ. Therefore,

$$\mathrm{R}_1,\ \rhd \to (\overleftarrow{\xi} \vee \overrightarrow{a\delta}) \vee \overrightarrow{\zeta}\ \vdash\ \rhd \to (\overleftarrow{\xi a} \vee \overrightarrow{\delta}) \vee \overrightarrow{\zeta}$$

by modus ponens. By induction hypothesis, we have

$$\mathrm{R}_1,\ \rhd \to (\overleftarrow{\xi a} \vee \overrightarrow{\delta}) \vee \overrightarrow{\zeta}\ \vdash\ \rhd \to \overleftarrow{\xi\beta} \vee \overrightarrow{\zeta}.$$

This completes the proof of the lemma. $\dashv$

Corollary 3.12. $\mathrm{R}_1,\ \rhd \to \overrightarrow{\xi} \vee \overrightarrow{\zeta}\ \vdash\ \rhd \to \overleftarrow{\xi} \vee \overrightarrow{\zeta}$, *for all nonempty words* $\xi, \zeta \in \mathcal{A}^*$.

Lemma 3.13. $\mathrm{R}_2,\ \rhd \to \overleftarrow{\xi} \vee \overrightarrow{\zeta}\ \vdash\ \rhd \to \overrightarrow{\xi\zeta}$, *for all nonempty words* $\xi, \zeta \in \mathcal{A}^*$.

Proof. By induction on $|\xi|$. If $|\xi| = 1$, then the formulas $\rhd \to \overleftarrow{\xi} \vee \overrightarrow{\zeta}$ and $\rhd \to \overrightarrow{\xi\zeta}$ are identical.

Now let $|\xi| \geq 2$, then $\xi = \beta a$ for a letter $a \in \mathcal{A}$ and a nonempty word β. Therefore,

$$\mathrm{R}_2,\ \rhd \to \overleftarrow{\beta a} \vee \overrightarrow{\zeta}\ \vdash\ \rhd \to \overleftarrow{\beta} \vee \overrightarrow{a\zeta}$$

by modus ponens. By induction hypothesis, we have

$$\mathrm{R}_2,\ \rhd \to \overleftarrow{\beta} \vee \overrightarrow{a\zeta}\ \vdash\ \rhd \to \overrightarrow{\xi\zeta}.$$

This completes the proof of the lemma. $\dashv$

Corollary 3.14. $\mathrm{R}_1,\ \mathrm{R}_2,\ \rhd \to \overrightarrow{\xi} \vee \overrightarrow{\zeta}\ \vdash\ \rhd \to \overrightarrow{\xi\zeta}$, *for all nonempty words* $\xi, \zeta \in \mathcal{A}^*$.

Lemma 3.15. *If* $\xi \stackrel{T}{\longmapsto} \zeta$ *then* $P_T,\ \rhd \to \overrightarrow{\xi}\ \vdash\ \rhd \to \overrightarrow{\zeta}$, *for all* $\xi, \zeta \in \mathcal{A}^*$.

Proof. Since T is applicable to ξ, we have $|\xi| \geq d$. Therefore, $\xi = a_i\alpha\beta$ and $\zeta = \beta\omega_i$, where $|\alpha| = d - 1$ and $|\beta| \geq 0$.
If $|\beta| = 0$, then

$$\begin{array}{rcll} P_T & \vdash & (\rhd \to \overrightarrow{\xi}) \to (\rhd \to \overrightarrow{\zeta}) & \text{by the axiom } (\mathrm{T}_2), \text{ and} \\ P_T,\ \rhd \to \overrightarrow{\xi} & \vdash & \rhd \to \overrightarrow{\zeta} & \text{by modus ponens.} \end{array}$$

Let $|\beta| > 0$, so

$$\begin{array}{rcll} P_T & \vdash & (\rhd \to \overrightarrow{\xi}) \to (\rhd \to \overrightarrow{\beta} \vee \overrightarrow{\omega_i}) & \text{by the axiom } (\mathrm{T}_1), \\ P_T,\ \rhd \to \overrightarrow{\xi} & \vdash & \rhd \to \overrightarrow{\beta} \vee \overrightarrow{\omega_i} & \text{by modus ponens,} \\ P_T,\ \rhd \to \overrightarrow{\xi} & \vdash & \rhd \to \overrightarrow{\zeta} & \text{by Corollary 3.14.} \end{array}$$

The lemma is proved. $\dashv$

Corollary 3.16. *If* $\xi \stackrel{T}{\Longmapsto} \zeta$ *then* $P_T,\ \rhd \to \overrightarrow{\xi}\ \vdash\ \rhd \to \overrightarrow{\zeta}$, *for all* $\xi, \zeta \in \mathcal{A}^*$.

The proof is trivial by definition of the tag system.

3.5. Production of the P_T-derivations. Here we show that the tag system T can produce P_T-derivations of the codes of words. As a preliminary let us introduce some notation and prove auxiliary lemmas.

Given $\alpha \in \mathcal{A}^*$, denote by P_{T,P_0} the set of axioms $P_T \cup \mathrm{H}$ and by $\mathsf{Code}_T(\alpha)$ the set of formulas:

$$\mathsf{Code}_T(\alpha) := \bigcup_{\beta \in \mathcal{A}^*,\ \alpha \overset{T}{\Longrightarrow} \beta} \mathsf{Code}(\beta).$$

It is clear that $\mathsf{Code}(\alpha) \subseteq \mathsf{Code}_T(\alpha)$ for all $\alpha \in \mathcal{A}^*$.

For any propositional calculus P denote by $\langle P \rangle$ the set of propositional formulas defined as follows:

$$\begin{aligned} \langle P \rangle := &\{B \mid A, A \to B \in P \text{ for some formula } A\} \cup \\ &\{\sigma A \mid A \in P \text{ and } \sigma \text{ is a substitution}\}. \end{aligned}$$

Furthermore, let $\langle P \rangle_0 = P$ and

$$\langle P \rangle_{n+1} = \langle \langle P \rangle_n \rangle$$

for $n \geq 0$. It follows easily that $\langle P \rangle_n \subseteq \langle P \rangle_{n+1}$ for all $n \geq 0$ and the set $[P]$ of all derivable formulas of the calculus P can be represented as

$$[P] = \langle P \rangle_\infty = \bigcup_{n \geq 0} \langle P \rangle_n .$$

Let A be a formula derivable from P. We say that A has the derivation height n, if $A \in \langle P \rangle_n$ and $A \notin \langle P \rangle_{n-1}$.

Consider the tag system T and the calculus P_{T,ω,P_0}. Let T halts on the input word ω, we take the minimal $n \geq 0$ such that $\langle P_{T,\omega,P_0} \rangle_n$ contains at least one code of some word $\alpha \in \mathcal{A}^*$ with $|\alpha| < d$:

$$N_\omega = \min\{n \geq 0 \mid \mathsf{Code}^*(\alpha) \cap \langle P_{T,\omega,P_0} \rangle_n \neq \emptyset, \text{ for some } \alpha \in \mathcal{A}^* \text{ with } |\alpha| < d\}.$$

If T does not halt, then we put $N_\xi = \infty$. Recall that $\mathsf{Code}^*(\alpha)$ is the set of all substitution instances of formulas in $\mathsf{Code}(\alpha)$.

Lemma 3.17. *$\langle P_{T,\omega,P_0} \rangle_{N_\omega} \subseteq \mathsf{Code}^*_T(\omega) \cup P^*_{T,P_0}$ for all $\omega \in \mathcal{A}^*$.*

Proof. We will prove by induction on $n \leq N_\omega$ that

$$\langle P_{T,\omega,P_0} \rangle_n \subseteq \mathsf{Code}^*_T(\omega) \cup P^*_{T,P_0}.$$

If $n = 0$, then $\langle P_{T,\omega,P_0} \rangle_0 = P_{T,\omega,P_0}$. It can easily be checked that the axiom W_ω is in $\mathsf{Code}^*_T(\omega)$ and all other axioms of P_{T,ω,P_0} are in P^*_{T,P_0}.

Let the induction assumption be satisfied for some $1 \leq n < N_\omega$. Since the right-hand side of the inclusion is closed under the substitution, we only consider the case of a formula B obtained by modus ponens from some formulas $A, A \to B \in \langle P_{T,\omega,P_0} \rangle_n$.

Since

$$\langle P_{T,\omega,P_0} \rangle_n \subseteq \mathsf{Code}^*_T(\omega) \cup P^*_{T,P_0},$$

we see by Lemma 3.8 that $A \to B$ is not a substitution instance of any formula of $\mathsf{Code}_T(\omega)$, i.e.

$$A \to B \notin \mathsf{Code}^*_T(\omega).$$

If A is a substitution instance of some axiom formula of P_{T,P_0}, then

$$A \to B \notin P^*_{T,P_0}$$

by Lemma 3.8. This contradicts the induction hypothesis. Hence

1. A is a substitution instance of the code for some word $\xi \in \mathcal{A}^*$ such that $\omega \overset{T}{\Longmapsto} \xi$, and
2. $A \to B$ is a substitution instance of some axiom of P_{T,P_0}.

Let us consider the following five cases:

Case 1. $A \to B$ is a substitution instance of the axiom T_1. Hence

$$A \in (\rhd \to \overrightarrow{a_i \alpha y})^*$$

for some letter $a_i \in \mathcal{A}$ and word $\alpha \in \mathcal{A}^*$ such that $|\alpha| = d-1$. Since the formula $A \in \mathsf{Code}^*(\xi)$, it is easily shown by Lemma 3.7 that

$$A \in (\rhd \to \overrightarrow{a_i \alpha \gamma})^*$$

for some nonempty word $\gamma \in \mathcal{A}^*$. Therefore B is the substitution instance of the code

$$\rhd \to \overrightarrow{\gamma} \vee \overrightarrow{\omega_i}$$

for the word $\zeta = \gamma\omega_i$ and $\xi \overset{T}{\longmapsto} \zeta$.

Case 2. $A \to B$ is a substitution instance of the axiom T_2. Hence

$$A \in (\rhd \to \overrightarrow{a_i \alpha})^*$$

for some letter $a_i \in \mathcal{A}$ and word $\alpha \in \mathcal{A}^*$ such that $|\alpha| = d-1$. Therefore B is the substitution instance of the code

$$\rhd \to \overrightarrow{\omega_i}$$

for the word $\zeta = \omega_i$ and $\xi \overset{T}{\longmapsto} \zeta$.

Case 3. $A \to B$ is a substitution instance of the axiom H. This case is impossible, since otherwise we would have $A \in (\rhd \to \overrightarrow{\alpha})^*$ for some $\alpha \in \mathcal{A}^*$, $0 < |\alpha| < d$. This contradicts to that

$$(\rhd \to \overrightarrow{\alpha})^* \cap \langle P_{T,\omega,P_0} \rangle_n \neq \emptyset$$

and $n < N_\omega$.

Case 4. $A \to B$ is a substitution instance of the axiom R_1. Hence

$$A \in (\rhd \to (y \vee \overrightarrow{az}) \vee u)^*$$

for some $a \in \mathcal{A}$. Since the formula $A \in \mathsf{Code}^*(\xi)$, we have by Lemma 3.7 that

$$A \in \left(\rhd \to \left(\overleftarrow{\xi_1} \vee \overrightarrow{a\xi_2}\right) \vee \overrightarrow{\xi_3}\right)^*$$

for some nonempty words $\xi_1, \xi_2, \xi_3 \in \mathcal{A}^*$. Therefore B is the substitution instance of the code

$$\rhd \to \left(\overleftarrow{\xi_1 a} \vee \overrightarrow{\xi_2}\right) \vee \overrightarrow{\xi_3}$$

for the word $\xi = \xi_1 a \xi_2 \xi_3$.

Case 5. $A \to B$ is a substitution instance of the axiom R_2. Hence

$$A \in (\rhd \to \overleftarrow{ya} \vee z)^*$$

for some $a \in \mathcal{A}$. Since the formula $A \in \mathsf{Code}^*(\xi)$, we have by Lemma 3.7 that

$$A \in \left(\rhd \to \overleftarrow{\xi_1 a} \vee \overrightarrow{\xi_2}\right)^*$$

for some nonempty words $\xi_1, \xi_2 \in \mathcal{A}^*$. Therefore B is the substitution instance of the code

$$\rhd \to \overleftarrow{\xi_1} \vee \overrightarrow{a\xi_2}$$

for the word $\xi = \xi_1 a \xi_2$.

Cases 1, 2, 3, 4, 5 exhaust all possibilities and so we have that $B \in \mathsf{Code}^*(\zeta)$ for some word $\zeta \in \mathcal{A}^*$ such that $\xi \overset{T}{\Longmapsto} \zeta$. Then $B \in \mathsf{Code}^*_T(\omega)$, since $\omega \overset{T}{\Longmapsto} \xi$ by induction hypothesis. The lemma is proved. ⊣

Now we prove that the code of each nonempty word on $\mathcal{A}$ derivable from P_{T,ω,P_0} with the derivation height less then or equal N_ω is the code of word produced from ω by the tag system T.

Corollary 3.18. *If* $\rhd \to \overrightarrow{\xi} \in \langle P_{T,\omega,P_0} \rangle_{N_\omega}$ *then* $\omega \overset{T}{\Longmapsto} \xi$, *for all* $\omega, \xi \in \mathcal{A}^*$.

Proof. By Lemma 3.17, we have

$$\langle P_{T,\omega,P_0} \rangle_{N_\omega} \subseteq \mathsf{Code}^*_T(\omega) \cup P^*_{T,P_0}.$$

Furthermore, the application of Lemma 3.8 yields

$$\mathsf{Code}^*_T(\omega) \cap P^*_{T,P_0} = \emptyset.$$

It is obvious that $\rhd \to \overrightarrow{\xi} \in \mathsf{Code}^*_T(\omega)$. Hence, $\omega \overset{T}{\Longmapsto} \xi$ by definition of the set $\mathsf{Code}^*_T(\omega)$. The lemma is proved. ⊣

§4. The proof of Theorem 2.6. Let us show that the following problem is undecidable: given a tag system $T = \langle \mathcal{A}, \mathcal{W}, d \rangle$ and a word $\omega \in \mathcal{A}$, determine whether $P_0 \leq P_{T,\omega,P_0}$.

Indeed, if the tag system T halts on the input word ω, then $\omega \overset{T}{\Longmapsto} \xi$ for some word $\xi \in \mathcal{A}^*$ such that $|\xi| < d$. Hence the code $\rhd \to \overrightarrow{\xi}$ of ξ is derivable from P_{T,ω,P_0} by Corollary 3.16. If we recall that P_{T,ω,P_0} contains the formula

$$(\rhd \to \overrightarrow{\xi}) \to A$$

for all $A \in P_0$, we obtain that $P_0 \leq P_{T,\omega,P_0}$.

Let $P_0 \leq P_{T,\omega,P_0}$, so by Lemma 3.6 we have

$$P_{T,\omega,P_0} \vdash \rhd \to \overrightarrow{\xi}$$

for some ξ such that $|\xi| < d$. By Corollary 3.18 we obtain $\omega \overset{T}{\Longmapsto} \xi$. Therefore, T halts on ω.

Thus, we reduce the halting problem of tag systems to the problem of recognizing extentions of the Σ-calculus P_0. Since the halting problem of tag systems is undecidable by Theorem 3.2, this completes the proof.

§5. The proof of Theorem 2.7. It is easy follows from Theorem 2.6 by Corollary 3.10.

§6. Acknowledgement. The author is grateful to Evgeny Zolin for useful comments and advices that improved the manuscript.

REFERENCES

[1] Grigoriy V. Bokov, *Completeness problem in the propositional calculus*, ***Intelligent Systems***, vol. 13 (2009), no. 1-4, pp. 165–182 (Russian).

[2] Martin Davis, ***Computability & unsolvability***, Information processing and computers, McGraw-Hill, 1958.

[3] M. D. Gladstone, *Some ways of constructing a propositional calculus of any required degree of unsolvability*, ***Transactions of the American Mathematical Society***, vol. 118 (1965), pp. 192–210.

[4] Kiyoshi Iséki, *On axiom systems of propositional calculi, xxiii*, ***Proc. Japan Acad.***, vol. 42 (1966), no. 8, pp. 875–877.

[5] Stephen Cole Kleene, ***Mathematical Logic***, Dover Publications, 2002.

[6] Alexander V. Kuznetsov, *Undecidability of the general problems of completeness, decidability and equivalence for propositional calculi*, ***Algebra and Logic***, vol. 2 (1963), no. 4, pp. 47–66 (Russian).

[7] Samuel Linial and Emil L. Post, *Recursive unsolvability of the deducibility, Tarski's comleteness, and independence of axioms problems of the propositional calculus*, ***Bulletin of the American Mathematical Society***, vol. 55 (1949), p. 50.

[8] Jan Łukasiewicz, *The shortest axiom of the implicational calculus of propositions*, ***Proceedings of the Royal Irish Academy. Section A: Mathematical and Physical Sciences***, vol. 52 (1948), pp. 25–33.

[9] Jerzy Marcinkowski, *A Horn clause that implies an undecidable set of Horn clauses*, ***Selected papers of the 7th Workshop on Computer Science Logic (CSL '93)***, vol. 832 (1994), pp. 223–237.

[10] Carew Meredith, *A single axiom of positive logic*, ***Journal of Computing Systems***, vol. 1 (1953), pp. 169–170.

[11] Marvin L. Minsky, *Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines*, ***Annals of Mathematics***, vol. 74 (1961), pp. 437–455.

[12] Emil L. Post, *Formal reduction of the general combinatorial decision problem*, ***American Journal of Mathematics***, vol. 65 (1943), pp. 197–215.

[13] Hourya Sinaceur, *Address at the Princeton University bicentennial conference on problems of mathematics (December 17–19, 1946), by Alfred Tarski.*, ***The Bulletin of Symbolic Logic***, vol. 6 (2000), no. 1, pp. 1–44.

[14] A. Tarski and J. Corcoran, ***Logic, semantics, metamathematics: Papers from 1923 to 1938***, Hackett Publishing Company, Incorporated, 1983.

[15] Hao Wang, *Tag systems and lag systems*, ***Mathematische Annalen***, vol. 152 (1963), no. 1, pp. 65–74.

[16] Mary Katherine Yntema, *A detailed argument for the Post-Linial theorems*, ***Notre Dame Journal of Formal Logic***, vol. 5 (1964), no. 1, pp. 37–50.

[17] J.J. Zeman, ***Modal Logic: The Lewis modal systems***, Clarendon Press, 1973.

[18] Evgeny Zolin, *Undecidability of the problem of recognizing axiomatizations of superintuitionistic propositional calculi*, ***Studia Logica***, (2013), pp. 1–19.

DEPARTMENT OF MATHEMATICAL THEORY OF INTELLIGENT SYSTEMS
LOMONOSOV MOSCOW STATE UNIVERSITY
MOSCOW, RUSSIAN FEDERATION
E-mail: bokovgrigoriy@gmail.com